Fourier-Net: Fast Image Registration with Band-Limited Deformation

Xi Jia\(^1\), Joseph Bartlett\(^1\)\(^2\), Wei Chen\(^1\), Siyang Song\(^3\), Tianyang Zhang\(^1\), Xinxing Cheng\(^1\), Wenqi Lu\(^4\), Zhaowen Qiu\(^5\)\(^6\), Jinming Duan\(^1\)\(^6\)†

\(^1\) School of Computer Science, University of Birmingham, UK
\(^2\) Department of Biomedical Engineering, University of Melbourne, Australia
\(^3\) Department of Computer Science and Technology, University of Cambridge, UK
\(^4\) Department of Computer Science, University of Warwick, UK
\(^5\) Institute of Information Computer Engineering, Northeast Forestry University, China
\(^6\) Alan Turing Institute, UK

Abstract

Unsupervised image registration commonly adopts U-Net style networks to predict dense displacement fields in the full-resolution spatial domain. For high-resolution volumetric image data, this process is however resource-intensive and time-consuming. To tackle this problem, we propose the Fourier-Net, replacing the expansive path in a U-Net style network with a parameter-free model-driven decoder. Specifically, instead of our Fourier-Net learning to output a full-resolution displacement field in the spatial domain, we learn its low-dimensional representation in a band-limited Fourier domain. This representation is then decoded by our devised model-driven decoder (consisting of a zero padding layer and an inverse discrete Fourier transform layer) to the dense, full-resolution displacement field in the spatial domain. These changes allow our unsupervised Fourier-Net to contain fewer parameters and computational operations, resulting in faster inference speeds. Fourier-Net is then evaluated on two public 3D brain datasets against various state-of-the-art approaches. For example, when compared to a recent transformer-based method, named TransMorph, our Fourier-Net, which only uses 2.2% of its parameters and 6.66% of the multiply-add operations, achieves a 0.5% higher Dice score and an 11.48 times faster inference speed. Code is available at https://github.com/xi-jia/Fourier-Net.

1 Introduction

Medical image registration aims to learn a spatial deformation that identifies the correspondence between a moving image and a fixed image, which is a fundamental step in many medical image analysis applications such as longitudinal studies, population modeling, and statistical atlases (Sotiras, Davatzikos, and Paragios 2013).

Iterative optimization techniques such as FFD (Rueckert et al. 1999), Demons (Vercauteren et al. 2009), ANTS (Avants et al. 2011), Flash (Zhang and Fletcher 2019) and ADMM (Thorley et al. 2021) have been applied to deformable image registration. However, such optimization-based approaches require elaborate hyper-parameter tuning for each image pair, and iteration towards an optimal deformation is very time-consuming, thus limiting their applications in real-time and large-scale volumetric registration.

† Corresponding author.
Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
ciency of such networks can be further improved by learning a low-dimensional representation of displacement field in a band-limited Fourier domain. Our arguments are based on our observation in Figure 1, where we notice it is sufficient to reconstruct an accurate full-resolution deformation (the third image of the second row in Figure 1) by using only a small number of coefficients in the band-limited Fourier domain. Inspired by this insight, we propose an end-to-end unsupervised approach that learns only a low-dimensional representation of displacement field in the band-limited Fourier domain. We term our approach the Fourier-Net.

By removing several layers in the expansive path of a U-Net style architecture, our Fourier-Net outputs only a small patch that stores low-frequency coefficients of displacement field in the Fourier domain. We then propose to directly apply a model-driven decoder to recover the full-resolution spatial displacement field from these low-frequency Fourier coefficients. This model-driven decoder contains a zero-padding layer that broadcasts complex-valued low-frequency signals into a full-resolution complex-valued map. The inverse discrete Fourier transform (iDFT) is then applied to this map to obtain the full-resolution spatial displacement field. Both zero-padding and iDFT layers are parameter-free and therefore fast. On top of Fourier-Net, we propose a diffeomorphic variant, termed Fourier-NetDiff. This network first estimates a stationary velocity field, followed by squaring and scaling layers (Dalca et al. 2018) to encourage the output deformation to be diffeomorphic.

2 Related Works

Unsupervised approaches can be based on either iterative optimization or learning. Iterative methods are prohibitively slow, especially when the images to be registered are of a high-dimensional form, such as 3D volumes. Over the past decades, many works have been proposed to accelerate such methods. (Ashburner 2007) used a stationary velocity field (SVF) representation (Legouhy et al. 2019), and proposed a fast algorithm DARTEL for image registration which computes the resulting deformation by using scaling and squaring from the SVF. Another fast approach for image registration is Demons (Vercauteren et al. 2009), which imposes smoothness on displacement fields by incorporating inexpensive Gaussian convolutions into its iterative process. Hernandez (Hernandez 2018) reformulated the Stokes-LDDMM variational problem used in (Mang and Biros 2015) in the domain of band-limited non-stationary vector fields and utilized GPUs to parallelize their methods. (Zhang and Fletcher 2019) developed the Fourier-approximated Lie algebras for shooting (Flash) for fast diffeomorphic image registration, where they proposed to speed up the solution of the Euler-Poincaré differential (EPDiff) equation used to compute deformations from velocity fields in the band-limited Fourier domain.

On the other hand, deep learning methods based on convolutional neural networks have been employed to overcome slow registration speeds. Among them, U-Net style networks have been proven to be an effective tool to learn deformations between pairwise images (Balakrishnan et al. 2019; Zhang 2018; Mok and Chung 2020b; Kim et al. 2021). While their registration performance is comparable with iterative methods, their inference can be orders of magnitude faster. RC-Net (Zhao et al. 2019) and VR-Net (Jia et al. 2021) cascaded multiple U-Net style networks to improve the registration performance, but their speed is relatively slow. Very recently, approaches, such as ViT-V-Net (Chen et al. 2021b) and TransMorph (Chen et al. 2021a), which combine vision transformers and U-Nets have achieved promising registration performance, but they involve much more computational operations and are therefore slow. Another group of network-based image registration techniques (De Vos et al. 2019; Qiu et al. 2021) is to estimate a grid of B-Spline control points with regular spacing, which is then interpolated based on cubic B-Spline basis functions (Rueckert et al. 1999; Duan et al. 2019). By estimating fewer control points, these networks perform fast predictions, but currently are less accurate.

Supervised approaches are also studied in medical image registration. However, they have several pitfalls: 1) it is generally hard to provide human-annotated ground truth deformations for supervision; and 2) if trained using numerical solutions of other iterative methods, the performance of these supervised registration methods may be limited by iterative methods. Yang et al. proposed Quicksilver (Yang et al. 2017) which is a supervised encoder-decoder network and trained using the initial momentum of LDDMM as the supervision signal. Wang et al. extended Flash (Zhang and Fletcher 2019) to DeepFlash (Wang and Zhang 2020) in a learning framework in lieu of iterative optimization. Compared to Flash, DeepFlash accelerates the computation of initial velocity fields but needs to solve a PDE (i.e., EPDiff equation) in the Fourier domain so as to recover the full-resolution deformation in the spatial domain, which can be slow. The fact that DeepFlash requires the numerical solutions of Flash (Zhang and Fletcher 2019) as training data attributes to lower registration performance than Flash.

Although DeepFlash also learns a low-dimensional band-limited representation, it differs from our Fourier-Net in four aspects, which we reckon our novel contributions to this area. First, DeepFlash is a supervised method that requires ground truth velocity fields calculated from Flash prior to training, whilst Fourier-Net is a simple and effective unsupervised method thanks to our proposed model-driven decoder. Second, DeepFlash is a multi-step method whose network’s output requires an additional PDE algorithm (Zhang and Fletcher 2019) to compute final full-resolution spatial deformations, whilst Fourier-Net is a holistic model that can be trained and used in an end-to-end manner. Third, DeepFlash needs two individual convolutional networks to estimate real and imaginary signals in the band-limited Fourier domain, whilst Fourier-Net uses only one single network directly mapping image pairs to a reduced-resolution displacement field without the need of complex-valued operations. Lastly, DeepFlash is essentially an extension of Flash and it is difficult for the method to benefit from vast amounts of data, whilst Fourier-Net is flexible and can easily learn from large-scale datasets. Due to these, our Fourier-Net outperforms DeepFlash (as well as Flash) by a significant margin in terms of both accuracy and speed.
3 Methodology

As illustrated in Figure 2, in Fourier-Net, the encoder first takes a pair of spatial images as input, and encodes them to a low-dimensional representation of displacement field (or velocity field if diffeomorphisms are imposed) in the band-limited Fourier domain. Then the decoder brings the displacement field (or velocity field) from the band-limited Fourier domain to the spatial domain, and ensures that they have the same spatial size as the input image pair. Next, the optional squaring and scaling layers are used to encourage a diffeomorphism in final deformations. Finally, by minimizing the loss function, an accurate deformation can be estimated, with which the warping layer deforms the moving image to be similar to the fixed image.

3.1 Encoder

The encoder aims to learn a displacement or velocity field in the band-limited Fourier domain. Intuitively, this may require convolutions to be able to handle complex-valued numbers. One may directly use complex convolutional networks (Trabelsi et al. 2017), they are suitable when both input and output are complex values, but complex-valued operations sacrifice computational efficiency. Instead, DeepFlash (Wang and Zhang 2020) tackles this problem by first converting input image pairs to the Fourier domain and then using two individual real-valued convolutional networks to learn the real and imaginary signals separately. Such an approach increases the training and inference cost (as listed in Table 1). Since our Fourier-Net estimates displacement fields in the band-limited Fourier domain from spatial images (inputs are real values but outputs are complex values), these approaches may not be well suited to our application.

To bridge the domain gap between real-valued spatial images and complex-valued band-limited displacement fields without increasing complexity, we propose to embed a DFT layer at the end of the convolutional network in the encoder. This is a simple and effective way to produce complex-valued band-limited displacement fields without the network being able to handle complex values itself. Let us denote the moving image as \( I_0 \), the fixed image as \( I_1 \), the convolutional network as CNN with the parameters \( \Theta \), the DFT as \( \mathcal{F} \), the full-resolution spatial displacement field as \( \phi \), and the complex band-limited displacement field as \( \mathbb{B}_\phi \). In this case, our encoder can be defined as \( \mathbb{B}_\phi = \mathcal{F}(\text{CNN}(I_0, I_1; \Theta)) \), resulting in a compact, efficient implementation as compared to DeepFlash and other complex convolutional networks. On the other hand, we also notice from our experiments (Table 1) that it is difficult to regress \( \mathbb{B}_\phi \) directly from \( I_0 \) and \( I_1 \) within a single CNN, i.e., \( \mathbb{B}_\phi = \text{CNN}(I_0, I_1; \Theta) \). We believe the reason being that: if the CNN directly learns a band-limited displacement field, it needs to go through two domains altogether: first mapping the spatial images to the spatial displacement field and then mapping this displacement field into its band-limited Fourier domain. In this case, the domain gap is too big. Our network however only needs to go through one domain and then DFT handles the second domain. By doing so, Fourier-Net is efficient and easy to learn. An illustration of this idea is given in Figure 3.

So far, we have given an intuitive explanation of how the encoder in our network learns. Here we discuss their mathematical relationship between the low-dimensional spatial displacement field \( \mathcal{S}_\phi = \text{CNN}(I_0, I_1; \Theta) \), its band-limited representation \( \mathbb{B}_\phi \), as well as the displacement field \( \phi \) (coming after the decoder) in the full-resolution spatial domain. For simplicity, we use a 2D displacement field as an example and the formulations below can be easily extended to 3D.
cases. A general discrete Fourier transform used on $\phi$ can be defined as follows:

$$\mathcal{F}(\phi)_{k,l} = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \phi_{i,j} e^{-\sqrt{-1}\pi \left( \frac{2\pi k}{M} i + \frac{2\pi l}{N} j \right)},$$  \hspace{1cm} (1)$$

where $\phi$ is of size $M \times N$, $i \in [0, M - 1]$ and $j \in [0, N - 1]$ are the discrete indices in the spatial domain, and $k \in [0, M - 1]$ and $l \in [0, N - 1]$ are the discrete indices in the frequency domain.

In our Fourier-Net, $\phi$ is actually a low-pass filtered displacement field. If we define a $M \times N$ sized sampling mask $\mathcal{D}$ whose entries are zeros if they are on the positions of high-frequency signals in $\phi$ and ones if they are on the low-frequency positions. With $\mathcal{D}$, we can recover the displacement field $\phi$ from Eq. (1)

$$\phi_{i,j} = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} \mathcal{D}_{k,l} [\mathcal{F}(\phi)]_{k,l} e^{\sqrt{-1}\pi \left( \frac{2\pi k}{M} + \frac{2\pi l}{N} \right)}. \hspace{1cm} (2)$$

If we shift all low-frequency signals of the displacement field to a center patch of size $\frac{M}{a} \times \frac{N}{b}$ ($\frac{M}{a}, \frac{N}{b}, a = 2Z, b = 2Z, Z, Z \in Z^+$), center-crop the patch (denoted by $\mathbb{B}_{\phi}$), and then perform the iDFT on this patch, we obtain $S_{\phi}$ in Eq. (3)

$$[S_{\phi}]_{\hat{i}, \hat{j}} = \frac{ab}{MN} \sum_{k=0}^{\frac{M}{a}-1} \sum_{l=0}^{\frac{N}{b}-1} [\mathbb{B}_{\phi}]_{\hat{i}, \hat{j}, \hat{k}, \hat{l}} e^{\sqrt{-1}\pi \left( \frac{2\pi \hat{k}}{M} + \frac{2\pi \hat{l}}{N} \right)}. \hspace{1cm} (3)$$

where $\hat{i} \in [0, \frac{M}{a} - 1]$ and $\hat{j} \in [0, \frac{N}{b} - 1]$ are the indices in the spatial domain, and $\hat{k} \in [0, \frac{M}{a} - 1]$ and $\hat{l} \in [0, \frac{N}{b} - 1]$ are the indices in the frequency domain. Note that $S_{\phi}$ is a low-dimensional spatial representation of $\phi$ and we are interested in their mathematical connection. Another note is that $S_{\phi}$ actually contains all the information of its band-limited Fourier coefficients in $\mathbb{B}_{\phi}$. As such, we do not need the network to learn the coefficients in $\mathbb{B}_{\phi}$ and instead only to learn its real-valued coefficients in $S_{\phi}$.

Since most of entries ($\frac{a \times b}{M \times N}$) in $\mathcal{F}(\phi)$ are zeros, and the values of rest entries are exactly the same as in $\mathbb{B}_{\phi}$, we can conclude that $S_{\phi}$ contains all the information $\phi$ can provide, and their mathematical connection is

$$[S_{\phi}]_{\hat{i}, \hat{j}} = ab \times [\phi]_{a \hat{i}, b \hat{j}} \hspace{1cm} (4)$$

With this derivation, we show that we can actually recover a low-dimensional spatial representation $S_{\phi}$ from its full-resolution spatial displacement field $\phi$, as long as they have the same low-frequency coefficients $\mathbb{B}_{\phi}$. This essentially proves that there exists a unique mapping function between $S_{\phi}$ and $\phi$ and that it is reasonable to use a network to learn $S_{\phi}$ directly from image pairs.

### 3.2 Model-Driven Decoder

The proposed decoder consists of a zero-padding layer, an iDFT layer, and an optional squaring and scaling module.

The output from the encoder is a band-limited representation $\mathbb{B}_{\phi}$. To recover the full-resolution displacement field $\phi$ in the spatial domain, we first pad the patch $\mathbb{B}_{\phi}$ containing mostly low-frequency signals to the original image resolution with zero values (i.e., $\mathcal{F}(\phi)$). We then feed the zero-padded complex-valued coefficients $\mathcal{F}(\phi)$ to an iDFT layer consisting of two steps: shifting the Fourier coefficients from centers to corners and then applying the standard iDFT to convert them into the spatial domain. The output from Fourier-Net is thus a full-resolution spatial displacement field. Both padding and iDFT layers are differentiable and therefore Fourier-Net can be optimized via standard back-propagation. We note that our proposed decoder is a parameter-free module that is driven by knowledge instead of learning and therefore fast.

We also propose a diffeomorphic variant of Fourier-Net which we term Fourier-Net-Diff. A diffeomorphic deformation is defined as a smooth and invertible deformation, and in Fourier-Net-Diff we need an extra squaring and squaring module for the purpose. The output of the iDFT layer can be regarded as a stationary velocity field denoted by $v$ instead of the displacement field $\phi$. In group theory, $v$ is a member of Lie algebra, and we can exponentiate this stationary velocity field (i.e., $\exp(v)$) to obtain a diffeomorphic deformation. In this paper, we use seven scaling and squaring layers (Ashburner 2007; Dalca et al. 2018) to impose such a diffeomorphism.

### 3.3 Warping Layer and Loss Functions

After the model-driven decoder, we obtain a full-resolution displacement field (or a diffeomorphic deformation) for the input image pair. We then deform the moving image using a warping layer to produce the warped moving image, which is then used to calculate the loss. We implement 2D and 3D spatial warping layers based on linear interpolation as in (Jaderberg et al. 2015) and (Balakrishnan et al. 2019).

We adopt an unsupervised loss which is computed from the moving image $I_1$, the fixed image $I_0$, and the predicted
displacement field $\phi$ or velocity field $v$. The training objective of our Fourier-Net is $\mathcal{L}(\Theta) = \min_\Theta \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{Sim}(I^0_i \circ (\phi_i(\Theta) + \text{Id}) - I^0_i) + \frac{1}{N} \sum_{i=1}^{N} \|\nabla \phi_i(\Theta)\|_2^2$, where $N$ is the number of training pairs, $\Theta$ is the network parameters to be learned, $\text{Id}$ is the identity grid, $\circ$ is the warping operator, and $\nabla$ is the first order gradient implemented using finite differences (Lu et al. 2016; Duan et al. 2016). The first term $\mathcal{L}_{Sim}$ defines the similarity between warped moving images and fixed images, and the second term defines the smoothness of displacement fields. Here $\lambda$ is a hyper-parameter balancing the two terms. As for Fourier-Net-Diff, the training loss is defined as $\mathcal{L}(\Theta) = \min_\Theta \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{Sim}(I^0_i \circ \text{Exp}(v_i(\Theta)) - I^0_i) + \frac{1}{N} \sum_{i=1}^{N} \|\nabla v_i(\Theta)\|_2^2$. $\mathcal{L}_{Sim}$ can be either mean squared error (MSE) or normalized cross-correlation (NCC), which we clarify in our experiments.

### 4 Experiments

#### 4.1 Datasets

**OASIS-1 dataset** (Marcus et al. 2007) consists of a cross-sectional collection of T1-weighted brain MRI scans from 416 subjects. In experiments, we use the pre-processed OASIS data\(^1\) provided by (Hoopes et al. 2021) to perform subject-to-subject brain registration, in which all 414 MRI scans are bias-corrected, skull-stripped, aligned, and cropped to the size of $160 \times 192 \times 224$. Automated segmentation masks from FreeSurfer are produced for evaluation of registration accuracy. This dataset also has 414 2D slices and marks extracted from their corresponding 3D volumes. We randomly split this 2D dataset into 201, 12, and 201 images for training, validation, and test. After pairing, we end up with 40200 (201 $\times$ 200), 22 (12-1), and 400 (201-1) image pairs for training, validation, and test, respectively.

**IXI dataset**\(^2\) contains nearly 600 MRI scans from healthy subjects. In experiments, we use the pre-processed IXI data provided by (Chen et al. 2021a) to perform atlas-based brain registration. The atlas is generated by (Kim et al. 2021). There are in total 576 $160 \times 192 \times 224$ 3D brain MRI volumes in this dataset. The dataset is split into 403 for training, 58 for validation, and 115 for testing. There is no pairing step for this dataset as it is an atlas-to-subject registration task.

#### 4.2 Implementation Details

We implement our Fourier-Net using PyTorch, where training is optimized using Adam with a fixed learning rate of 0.0001. We tune built-in hyper-parameters on a held-out validation set. Specifically, we use MSE to train both 2D and 3D OASIS for 10 and 1000 epochs, respectively, and $\lambda$ in $\mathcal{L}(\Theta)$ is set to 0.01. For 3D OASIS, an additional Dice loss is used with its weight being set to 1. On 3D IXI, we train the models with NCC loss for 1000 epochs with $\lambda = 5$. All deep models are trained with an Nvidia A100 GPU.

The CNN in Fourier-Net has 6 convolutional blocks. The initial 4 blocks contain 2 convolutional layers in each block. The first layer maintains the same spatial resolution as inputs, while the second layer performs a downsampling with $2 \times 2$. For this dataset as it is an atlas-to-subject registration task. The dataset is split into 403 for training, 58 for validation, and 115 for testing. There is no pairing step for this dataset as it is an atlas-to-subject registration task.

#### 4.3 Ablation Studies and Parameter Tuning

The first question we ask is what is the most suitable resolution (i.e., patch size) of a band-limited displacement field is? A very small patch will rapidly decrease model parameters as well as training and inference time but may lead to lower performance. A very large patch could retain registration accuracy but may increase training and inference time, thus eliminating the advantages of our method.

In Table 1, we use 2D OASIS images for ablation studies and investigate the impact of different patch sizes, i.e., $20 \times 24$ and $40 \times 48$, which are respectively $\frac{1}{4}$ and $\frac{1}{4}$ of the original image size. It can be seen that the $40 \times 48$ patch improves Dice by 2% over the $20 \times 24$ patch, with only a slight difference in mult-adds operations. The Dice score of our Fourier-Net ($40 \times 48$) is already close to the full-resolution U-Net backbone (last row in this Table), which however has 2.5 times mult-adds costs than our Fourier-Net ($40 \times 48$).

We also prove the necessity of embedding a DFT layer in the encoder. Without this layer, our encoder is purely a CNN that has to learn complex-valued Fourier coefficients from image pairs. Following DeepFlash (Wang and Zhang 2020), we use two networks to separately compute the real and imaginary parts of these complex coefficients. As re-

| Patch DFT SS Dice$^\dagger$ $|J|_{<0\%}$ MA(M) |
|---|---|---|---|---|
| $20 \times 24$ | ✗ | ✗ | $0.664 \pm 0.040$ | $0.158 \pm 0.206$ | 891 |
| $20 \times 24$ | ✓ | ✗ | $0.732 \pm 0.042$ | $0.434 \pm 0.355$ | 679 |
| $40 \times 48$ | ✗ | ✗ | $0.735 \pm 0.037$ | $0.0 \pm 0.0$ | 679 |
| $40 \times 48$ | ✓ | ✗ | $0.675 \pm 0.038$ | $0.279 \pm 0.257$ | 1310 |
| $40 \times 48$ | ✓ | ✓ | $0.756 \pm 0.039$ | $0.753 \pm 0.407$ | 888 |
| U-Net | ✓ | ✓ | $0.756 \pm 0.037$ | $<0.0001$ | 888 |

\(1\)https://learn2reg.grand-challenge.org/Learn2Reg2021/

\(2\)https://brain-development.org/ixi-dataset/
We manage to compile and run Flash (Zhang and Fletcher 2019), DeepFlash (Wang and Zhang 2020), and B-Spline-Diff (Qiu et al. 2021). We manage to compile and run Flash in CPU, but its official GPU version keeps throwing segmentation fault errors. We report the performance of Flash on three band-limited patch sizes, and its built-in hyper-parameters are grid-searched over 252 different combinations on the whole validation set for each size. We also manage to run Deep-Flash with supervision from Flash’s results. We train DeepFlash on all 40200 training pairs for 1000 epochs with more than 40 different combinations of hyper-parameters and report the best results. B-Spline-Diff is also trained with all training pairs using its official implementation.

In Table 2, all Fourier-Net variants outperform competing methods in terms of Dice. Specifically, our Fourier-Net achieves a 0.748 Dice score with 0.007 seconds inference speed per image pair. Compared to Flash using a 40×48 patch, Fourier-Net improves Dice by 1.5% and is 12,257 times faster. Though DeepFlash is much faster than Flash, we find that DeepFlash is very difficult to converge and as such achieves the lowest Dice score (0.597). Moreover, DeepFlash is not an end-to-end method, because its output (band-limited velocity field) requires an additional PDE algorithm to compute the final deformation. As such, it is much slower than deep learning methods such as ours or B-Spline-Diff (0.012 seconds per image pair on CPU). Note that the computational time is averaged on the whole test set, including the cost of loading models and images.

Note that the speed advantage of Fourier-Net on CPU decreases when we use larger models such as Fourier-Net4, but its performance can be boosted by 1.1% compared to Fourier-Net1 in terms of Dice.

We also list the percentage of negative values of Jacobian determinant of deformation for all compared methods in Table 2. Though both Flash and B-Spline-Diff are diffeomorphic approaches, neither of them produces perfect diffeomorphic deformations on this dataset. The proposed three Fourier-Net-Diff variants, however, barely generate negative Jacobian determinants and are therefore diffeomorphic.

### 3D OASIS
In Table 3, we further compare Fourier-Net with other methods on the MICCAI 2021 Learn2reg challenge dataset. Though Fourier-Net is slightly lower than LapIRN (Mok and Chung 2020a) in Dice, it achieves a better Hausdorff distance than LapIRN with a 0.059 improvement. If we use a larger Fourier-Net4, it can achieve the lowest HD95, suggesting that Fourier-Net is able to obtain comparable results on par with state-of-the-art on this dataset.

### 4.5 Comparison on Atlas-Based Registration

#### 3D IXI
In Table 4, we first compare our Fourier-Net with iterative methods such as Flash and deedsBCV (Heinrich, Maier, and Handels 2015) and deep learning methods such as TransMorph-B-Spline (Chen et al. 2021a), which is a combination of TransMorph and B-Spline-Diff. Note that for Flash, 200 combinations of hyper-parameters are grid-searched using 5 randomly selected validation samples. We do not include all images in the validation set because tuning Flash on CPU can take up to 30 minutes per pair.

The proposed Fourier-Net achieves the highest Dice score (0.763) with 4.42s inference speed per image pair. By using less number of kernels in each layer, Fourier-Net achieves the highest speed (1.92s) on CPU, which is faster than all other deep learning methods, while retaining a competitive accuracy. Furthermore, Fourier-Net4 outperforms...
<table>
<thead>
<tr>
<th>Methods</th>
<th>Dice†</th>
<th></th>
<th>J</th>
<th>&lt;0.0%</th>
<th>Parameters</th>
<th>Mult-Adds (G)</th>
<th>CPU (s)</th>
<th>GPU (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affine</td>
<td>.386±.195</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SyN* (Avants et al. 2011)</td>
<td>.645±.152</td>
<td>&lt;0.0001</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>NiftyReg* (Modat et al. 2010)</td>
<td>.645±.167</td>
<td>&lt;0.0001</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>LDDMM* (Beg et al. 2005)</td>
<td>.680±.135</td>
<td>&lt;0.0001</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Flash (Zhang and Fletcher 2019)</td>
<td>.692±.140</td>
<td>0.0±0.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>deedsBCV*</td>
<td>.733±.126</td>
<td>0.147±0.050</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>VoxelMorph-1* (Balakrishnan et al. 2019)</td>
<td>.728±.129</td>
<td>1.590±0.339</td>
<td>274,387</td>
<td>304.05</td>
<td>9.373</td>
<td>0.391</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VoxelMorph-2* (Balakrishnan et al. 2019)</td>
<td>.732±.123</td>
<td>1.522±0.336</td>
<td>301,411</td>
<td>398.81</td>
<td>10.530</td>
<td>0.441</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VoxelMorph-Diff*</td>
<td>.580±.165</td>
<td>&lt;0.0001</td>
<td>307,878</td>
<td>89.67</td>
<td>3.691</td>
<td>0.418</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B-Spline-Diff* (Qu et al. 2021)</td>
<td>.742±.128</td>
<td>&lt;0.0001</td>
<td>266,387</td>
<td>47.05</td>
<td>7.076</td>
<td>0.437</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TransMorph* (Chen et al. 2021a)</td>
<td>.754±.124</td>
<td>1.579±0.328</td>
<td>46,771,251</td>
<td>657.64</td>
<td>22.035</td>
<td>0.443</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TransMorph-Diff* (Chen et al. 2021a)</td>
<td>.594±.163</td>
<td>&lt;0.0001</td>
<td>46,557,414</td>
<td>252.61</td>
<td>10.389</td>
<td>0.438</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TransMorph-B-Spline* (Chen et al. 2021a)</td>
<td>.761±.122</td>
<td>&lt;0.0001</td>
<td>46,806,307</td>
<td>425.95</td>
<td>18.138</td>
<td>0.442</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fourier-NetS</td>
<td>.759±.132</td>
<td>0.009±0.008</td>
<td>1,050,800</td>
<td>43.82</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fourier-Net-DiffS</td>
<td>.756±.130</td>
<td>0.0±0.0</td>
<td>1,050,800</td>
<td>43.82</td>
<td>6.202</td>
<td>0.332</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fourier-Net</td>
<td>.763±.129</td>
<td>0.024±0.019</td>
<td>4,198,352</td>
<td>169.07</td>
<td>4.423</td>
<td>0.342</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fourier-Net-Diff</td>
<td>.761±.131</td>
<td>0.0±0.0</td>
<td>4,198,352</td>
<td>169.07</td>
<td>8.679</td>
<td>0.345</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Performance comparison between different methods on IXI. Results of the methods labeled with * are taken from TransMorph (Chen et al. 2021a), as we used the exact same data splitting and testing protocol as TransMorph. The reported runtimes of all deep methods are computed by us on the same machine and are averaged on the whole testing set.

TransMorph by 0.5% in Dice with only 2.2% of its parameters and 6.66% of its multi-adds. In terms of inference speed, Fourier-NetS is 11.48 times faster than TransMorph (22.035 seconds) on CPU. Finally, Table 4 (3rd column) indicates that Fourier-Net-Diff barely generates any folding and thus effectively preserves diffeomorphisms.

From Figure 4, we can observe that Flash’s deformation also has no foldings, but it over-smoothes its displacement field, resulting in a less accurate warping. Figure 4 (last row) shows that only Flash and Fourier-Net produce strictly band-limited Fourier coefficients, and that the deformation of Fourier-Net-Diff is no longer band-limited due to the use of the squaring and scaling layers.

5 Conclusion

In this paper, we propose to learn a low-dimensional representation of displacement/velocity field in the band-limited Fourier domain. Experimental results on two brain datasets show that our Fourier-Net is more efficient than state-of-the-art methods in terms of speed while retaining a comparative performance in terms of registration accuracy.
Acknowledgements

The computations described in this research were performed using the Baskerville Tier 2 HPC service. Baskerville was funded by the EPSRC and UKRI (EP/T022221/1 and EP/W032244/1) and is operated by Advanced Research Computing at the University of Birmingham. Xi Jia is partially supported by the Chinese Scholarship Council.

References


